Assessing Transit Access to Ramsey County Service Facilities

Prepared by:

Ben Tomhave in GIS 5578: GIS Programming
Instructor: David Haynes, Department of Geography, Environment & Society
----
College of Liberal Arts

Prepared in Collaboration with:

Max Holdhusen
Senior Policy Analyst,
Policy & Planning Department,
Ramsey County

RAMSEY COUNTY

Resilient Communities Project
University of Minnesota
Building community-university partnerships for sustainability

The project on which this report is based was completed in collaboration with Ramsey County as part of the 2018–2019 Resilient Communities Project (RCP) partnership. RCP is a program at the University of Minnesota's Center for Urban and Regional Affairs (CURA) that connects University faculty and students with Minnesota communities to address strategic projects that advance local resilience and sustainability.

Resilient Communities Project
University of Minnesota
330 HHHSPA
301—19th Avenue South
Minneapolis, Minnesota 55455
Phone: (612) 625-7501
E-mail: rcp@umn.edu
Web site: http://www.rcp.umn.edu

Resilient Communities Project
University of Minnesota
Building Community-University Partnerships for Resilience

TABLE OF CONTENTS

Ben Tomhave GIS 5578
14 December 2018

ASSESSING MIDDAY TRANSIT ACCESS TO RAMSEY COUNTY SERVICE FACILITIES

INTRODUCTION:

In their capacity as administrative agencies of the state, counties provide many areas of government support. These services range from social services and corrections to public health and library facilities. Serving the eastern twin cities region, Ramsey County provides the aforementioned services as well as many more. In serving residents of the county, Ramsey County officials and staff seek to foster a "vibrant community where all are valued and thrive...and [equitable] access to opportunity and mobility for all residents is enhanced."

Many of the services provided by the county are utilized by those that are economically disadvantaged. Many who rely most heavily on county facilities cannot afford to own a personal car and instead access county facilities by transit, with most trips occurring during midday hours between 10am and 2pm. During this time of day, however, many transit routes either do not run or have much less frequent service, further hampering residents' mobility and their ability to access important public services. In response to this issue of access, Ramsey County, established a priority in 2017 to identify locations for effective and equitable service facilities. This report evaluates the current transit accessibility of Ramsey County Corrections facilities and provides resources that can help determine future facility locations that would offer the most midday transit accessibility to the greatest number of Ramsey County residents.

DATASET DESCRIPTION:

Several sources of data are necessary to accurately address this problem. The data utilized in this project can be split into three aggregate categories: transit data, facility/client data, and county information. Transit data for this project was extracted from General Transit Feed Specifics (GTFS)[1] and bus headway data.[2] GTFS data is the standard format for transit data and is comprised of several text files that together create the transit network. These text files, for example, contain information related to the stop and route locations as well as the schedule for each

---

[1] Obtained from: https://transitfeeds.com/p/metro-transit/179
[2] Obtained from: https://gisdata.mn.gov/dataset/us-mn-state-metc-trans-transit-count-headway-rt

bus. While GTFS data outlines the transit network, the bus headway data provides aggregate statistics for each segment of road within the network. In particular, this dataset provides the number of buses for each route that travel along a road segment, down to the street block level, as well as each route's headway, defined as the number of minutes between consecutive buses of one route. Thus, headway data details the frequency of a route, while GTFS data outlines the physical locations of the transit network.

In order to tailor this transit accessibility to the county facilities, client location information was also needed. This information was provided by county employees and consisted of the home address of each client as well as the facility that the client used. The last pieces of data acquired for this research were the census tract[3] and county boundary shapefiles.[4]

METHODOLOGY:

The first step in addressing the problem statement defined above was filtering and manipulating that data into a usable format. In particular, the client locations provided by the county needed to be converted to latitude and longitude coordinates. In order to perform this conversion, an ArcGIS connection to Ramsey County's server was enabled. After enabling this connection, the county's publicly accessible geocoder[5] was added to ArcPro's list of address locators and used as the geocoder in ArcPro's Geocode Addresses tool. Following the completion of this tool, the X and Y coordinates of each client's address were added to the resulting geocoded table of addresses using the AddXY_management tool. The final step in filtering the client addresses was to export only the locations with the best match to a CSV file. In addition to filtering the client locations, transit data also needed to be filtered. In order to simplify this process, transit data was first uploaded to a postgresql database from which a query for all routes associated with each stop was run. This file was saved in the input data folder as "routesPerStop.csv."

Following the creation of the filtered input files, the initial step of the Python 3 script could be run. To create the requisite shapefiles for display in ArcPro, the CSV's corresponding to the geocoded client location and route geometry described above were input into a self- created function which employed arcpy's MakeXYEventLayer_management and FeatureClassToShapefile_conversion functions. After specifying the CSV columns for each file that correspond to the X coordinates and Y coordinates a shapefile was output and added to the project geodatabase. The

[3] Obtained from: https://gisdata.mn.gov/dataset/us-mn-state-metc-society-census2010population
[4] Obtained from: https://gisdata.mn.gov/dataset/us-mn-co-ramsey-bdry-admin-boundary-data
[5] https://maps.co.ramsey.mn.us/arcgis/rest/services/Services/RCGeocoder_Composite/Geocod eServer

immediate problem that stemmed from creating these shapefiles, however, was that they spanned the entire twin cities region rather than just the extent of Ramsey County. To address this problem of excess data, a shapefile corresponding to the extent of the county was added to the geodatabase using FeatureClassToGeodatabase and used as the clipping feature in arcpy's Clip_analysis. By using this consistent clipping feature, and iterating through all other shapefiles (Census Tract, Client Location, and Route Geometry) as the input to the Clip_Analysis tool, new shapefiles were created that only contained the data within Ramsey County.

Using these clipped shapefiles, the next step of the model was to create a table that contained all transit stops and routes that were near to a residents' home location. In particular, a near table was created using arcpy's GenerateNearTable function and populated with the nearest 20 bus stops within a quarter mile of each client. In order to visualize this table prior to running other portions of the code, the pandas dataframe function was loaded and used to create a dataframe where each column corresponded to a field name from the GenerateNearTable output. In a self-defined function, this dataframe was populated using an arcpy search cursor that iteratively searched through the near table and appended each row's cell value to the corresponding column. While the near table provides valuable stop-specific information, it only contains the client's ID number and the nearest 20 transit stop IDs and distances. To obtain more complete information such as each client's location rather than just their ID number, a query table was formed using the MakeQueryTable Arcpy function. This query table was populated with the records in which the Client and Stop ID from the near table both matched with the Client and Stop ID's from the clipped client and stop location shapefiles.

The result of this process was one table that contained all stop and client information for each of the 20 nearest bus stops to each client. A sample portion of the table is shown below in Table 1 where select client ID's are matched with the nearest stops to them and four of the routes associated with each stop. Client latitude and longitude coordinates are removed for the sake of privacy. The total size of this table is 40,291 rows by 30 columns.

ClipAddress_clientid	ClipStopsRou_stop_id	ClipStopsRou_BUS1	ClipStopsRou_BUS2	ClipStopsRou_BUS3	ClipStopsRou_BUS4
140555	49450	16	262	3	62
140555	11912	16	262	3	62
137322	49450	16	262	3	62
130861	20036	141	25	4	801
79263	20036	141	25	4	801
141157	20036	141	25	4	801
142391	52532	223	264	32	801
131363	49398	21	417	53	61

Table 1: Sample Snippet of Client-Route Relationship Table for The Nearest Four Routes

Following the creation of this complete table of information, each transit route was subsequently ranked by the number of residents that were near the route. This ranking was performed by utilizing self-created functions based around the CSV and collections modules. First, a dictionary was created for each user in which the user ID was the key and the nearby routes associated with this route, from the near table process described above, were the associated values. Then using the defaultdict function of the collections module, a dictionary was created where each unique route was added to a master list. Using the chain function of the itertools module, an iterable for each route was returned over which the counter function of the collections module iterates and counts and then ranks the number of unique times a route is within 0.25 miles of a client.

Next, a new field (column of the dataframe) titled ‘ProximityRank’ was created in the clipped route shapefile where the new field value for each route was equal to the number of residents near the route as contained within the aforementioned route proximity list. Thus, routes that were close to a larger number of residents had a larger value ProximityRank. To best convey the transit accessibility of any region in the county, however, it was determined that ranking the transit lines by their proximity to users was an adequate first step but did not fully capture the loss of transit accessibility during midday hours. Consequently, a second new field labeled ‘routeProxandFreq’ was created using the arcpy management function “Add Field” and then filled using the Calculated Field function. The field was calculated by multiplying the ProximityRank for each route by that route’s midday bus count, as obtained from the headway dataset (described in the dataset section above). By following this method, routes that were near many residents but had low or no midday service, in turn, had a low accessibility rating while routes that were near many residents and had high midday frequency had large accessibility measures.

Finally, the clipped client location shapefile containing the new fields was added to an ArcPro map along with the clipped route features, and census tracts. Before displaying the data, one final aggregation was performed by using the arcpy Summarize Fields and Summary Statistics functions in combination to create a final new field that was the sum of each routes’ average accessibility measure within a census tract. This last step was taken in order to convey the cumulative accessibility of an area serviced by multiple transit routes. The symbology of the census tract layer was then updated to be a graduated color scale based on the cumulative transit accessibility within the census tract. Finally, a heat map of the residents’ home locations was added above the census tract layer to make the connection between transit accessibility and client location even more clear.

RESULTS & DATA ANALYSIS

The final map of midday transit accessibility to Ramsey County corrections facilities is shown below in Figure 1 where the transit accessibility is aggregated by census tract and the three corrections facilities are depicted as the three building icons. From this figure, it is apparent that the northeast corner of Ramsey County near White Bear Lake has no midday transit service. In addition, while most of downtown St. Paul is well served by transit, there are several pockets in which a census tract either has no residents living within it or there are no stops within the census tract served by midday routes. Regarding the corrections facility placement, however, all three appear to be situated in areas of relatively high transit accessibility.

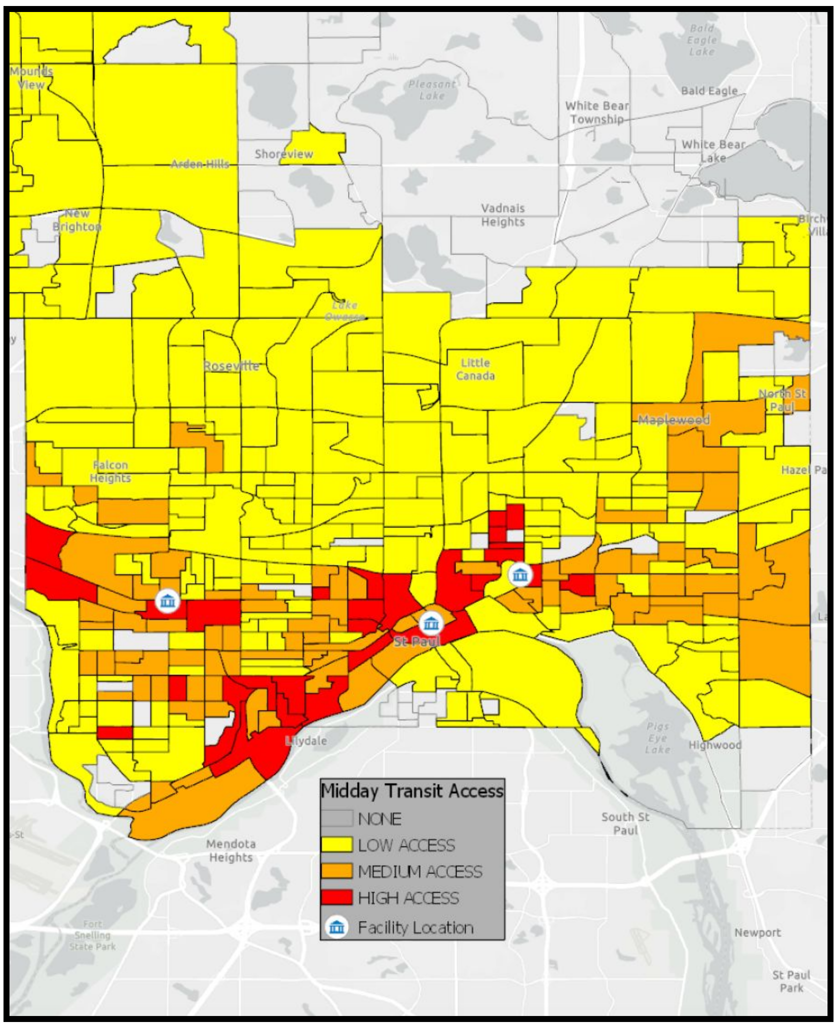


Figure 1: Midday Transit Accessibility to Ramsey County Corrections Facilities

When looking at Figure 2, however, more detail regarding the transit accessibility can be gained. In this figure, the access value is used to color code the census tract boundaries rather than their interior. This step was taken in order to visualize the heat map of where corrections facility residents live. With this added detail, the issue behind the lack of service in the White Bear Lake region is apparent given that a fairly large number of corrections residents live in this part of the county. Additionally, this figure demonstrates that if the central corrections facility were moved slightly to the northwest, it would be located in a region of higher client density and transit associability.

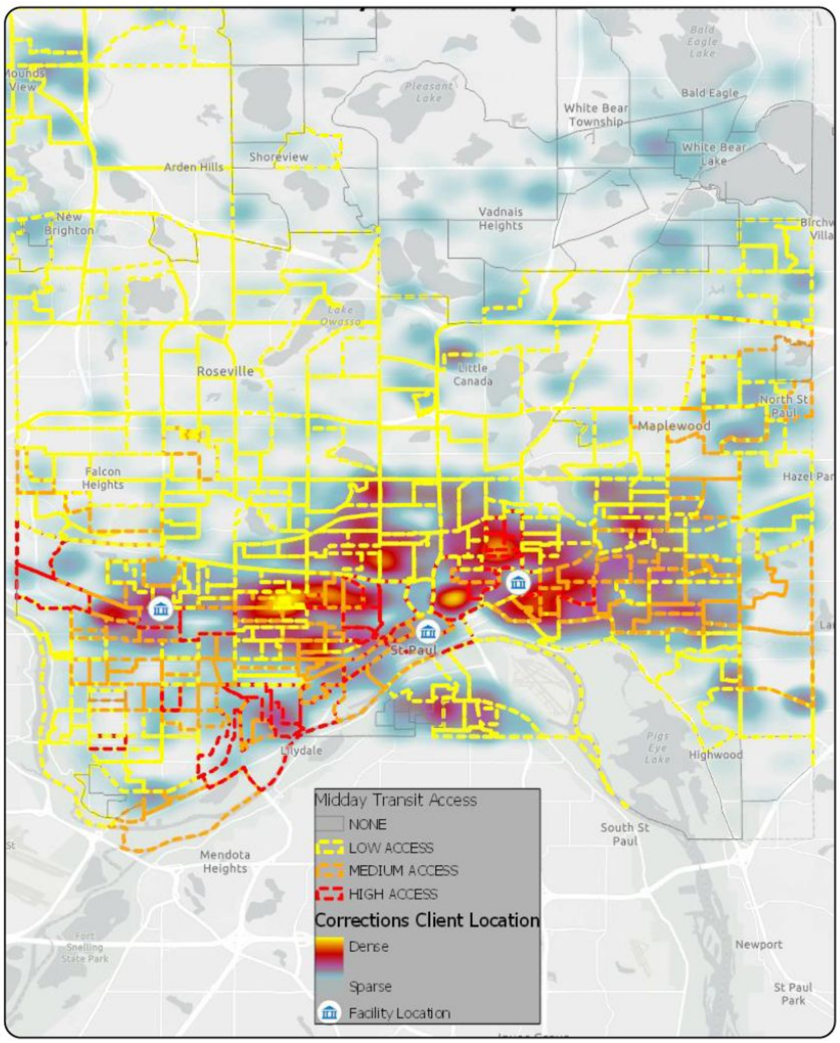


Figure 2: Midday Corrections Facility Transit Access with Client Location Heat Map

CHALLENGES & PROBLEMS:

In developing the python code necessary to produce the results shown above, several challenges were faced. The primary challenge faced in this project centered around creating the input datafiles that the python script worked with. The reason this step was particularly challenging was in part due to the number of datasets that needed to be condensed into one input file. For example, in creating an input file that related each stop to the routes that serviced the stop, three separate GTFS text files had to be joined to one another with varying query statements which often conflicted with each other and resulted in errors when attempting to do this stage of the project in ArcGIS. This problem was resolved by creating all transit input files from a database because the query interface and functionality within the database program was much smoother and quicker than in ArcGIS.

A second problem that was overcome in this project was working with input client data that was given in the form of an address and needed to be converted to latitude and longitude coordinates. A self-defined script was initially unable to be created as the ArcPro default “Address Locator” required a certain amount of purchased credits to be used. After discussing this problem with Ramsey County, however, the problem was resolved by adding a server extension in ArcPro that connected to Ramsey County’s publicly available geocoder which was then used to convert all the client location information into the necessary latitude-longitude format.

CONCLUSION & NEXT STEPS:

From the results of this accessibility model, it is apparent that while the midday transit access is inadequate for northwestern Ramsey County residents, the corrections facilities, for the most part, are situated in areas of the county that maximize their transit accessibility. In the future, this project can further be refined in several ways. First, the definition of transit accessibility can be further restricted to the accessibility residents have to a particular corrections facility rather than any one of the three possible facilities. Additionally, the script created for the corrections facility dataset can be applied to other county facilities by simply changing the input file of client locations to provide a more complete picture of the current state of transit accessibility to county facilities.

Finally, if an ArcGIS toolbox were created to encompass the features of this script, the value of this script could be immensely increased as county officials unfamiliar with python syntax could quickly run the tool to obtain insight into the accessibility attributes of a particular facility. Overall, this model is an effective first step in assessing the current level of transit access to county facilities and can be used in the future in helping to inform decisions regarding facility relocation in order to ensure that the largest possible number of residents are served by the most frequent and accessible transit service.

APPENDIX: CODE EXPLAINED

INTRODUCTION:

In order to investigate the midday transit accessibility of Ramsey County facilities, a python code was created to expedite the process and will be described below. The general idea of the code was to encapsulate both clients' proximity to a given route and the frequency of the route. Thus the transit access (per census tract) was determined as the sum of all the individual transit routes' access measure within that census tract where a route's access measure was calculated as the number of people within 0.25 miles of a specific route multiplied by the number of that route's buses passing over a give stretch of the road within the midday period of time defined by Metro Transit to be 10:00 a.m.- 2:00 p.m. It should thus be noted that this access measure cannot be translated to a tangible and physical quantity but should instead be used as a qualitative measure to compare access in different areas of the county.

In addition to the census tract level aggregated access measures described above, this project also created maps of the cumulative midday transit service on all roads in Ramsey County. This was calculated by summing the count of all Metro Transit buses that passed over a given stretch of road during the midday hours. Given that this was a simple aggregation of the midday counts by road segment the code has not been attached and this measure will not be discussed further. Additionally, the plots at the end of this report visualize only the bus transit access as it is already known that the Green Line light rail has high frequency service throughout the midday period.

Census Tract Level Access Code Explained:

This code is contained within the file ***RCPOverview.py*** which was created for the corrections dataset but with some minor adjustments, this code should also be usable for other facilities. It should be noted that in order to run this python code, the arcpy and Fiona libraries/ packages need to be installed. In addition, both the file ***RCPOverview.py*** and ***FileNameAndLocationGeneration.py*** must be located in the same directory (folder) in order to run properly.

Additionally, the user provided files that are necessary to run this program are:

- ***Geocoded Client Dataset (in csv format)***
- ***routesPerStop.csv (This was included in the zipped file with this code)***
    - CSV file detailing all routes that service a particular transit stop.
- ***County Boundary & Census Tract Shapefile (Included in zipped folder)***
    - Shapefile containing county extent and Census Tracts
- ***routeShapeXYWKday Shapefile(Included in zipped folder)***
    - Shapefile detailing the weekday route geometry

1. MODULE IMPORT & FUNCTION DEFINITION

- The first portion of this code imports the necessary modules [arpy, os, pandas, as pd, collections,chain from itertools, fiona, and datetime from datetime] in addition to defining several user-created functions that are utilized in the body of the code
- These functions should remain the same for all facilities with the exception of the ***getQueryTable*** function in which the ***inFieldString*** parameter may need to be changed in order to match the field names of the input data from the client location file.

2. **SET INPUT DATA LOCATION AND FILE NAMES**
   - In order to set this information there are two options, either the user can run the code as is and he or she will be presented with prompts to follow, or the user can input the data manually
   - If the first option is chosen, the ***exec(open())*** command will be activated which runs the file ***FileNameAndLocationGeneration.py***. The user will then be presented with several yes/no questions in order to determine where the correct files are located. Before running the code, the user should know where the following folders are located
      i. ***InputData folder***
      ii. ***Geodatabase (Output) folder***
      iii. ***County Boundary Shapefile folder***
      iv. ***Census Tract folder***
      v. ***Query Output Folder (Must be different that geodatabase folder)***
   - A sample of what this process would look like is shown in **Figure 1** below where the user input is highlighted in yellow

```
Enter File Path To Input Data Exactly With No Quotes: (Copy-paste directly
from file explorer)>>C:\rcpInputFinalData

Is client location information stored in a csv file? (Y or N)>>y

Is client location data stored in the folder C:\rcpInputFinalData? (Y or
N)>>y

************************************
File Number 0: matchedAddressSimplified.csv
File Number 1: routeShapeXY.csv
File Number 2: routeShapeXYWKday.csv
File Number 3: routesPerStop.csv

Which file number of the above choices [range(0, 3)] is the client location
csv? >>0
Client Location Csv File Selected As: matchedAddressSimplified.csv

************************************
File Number 0: matchedAddressSimplified.csv
File Number 1: routeShapeXY.csv
File Number 2: routeShapeXYWKday.csv
File Number 3: routesPerStop.csv

Which file number of the above choices [range(0, 3)] is the routesPerStop
information csv? >>3
Route Information Csv File Selected As: routesPerStop.csv

************************************

Please Specify Name For Geodatabase (OutputFileLocation) without file
extension.
Name must start with a character and be less than 10 characters.
e.x)[WFSOuput] >>TestOutput
```

Figure 1: Sample User Input Prompts (with input highlighted in yellow)

   - If the user instead choses to input all the paths and file names directly the *exec(open())* should be commented out and all the lines shown in Figure 2 should be un-commented and filled out with proper formatting in place of the exclamation marks which are currently used as placeholders.

```
#gdbPath=!
#gdbName=!
#inputDataPath=!
#queryDataPath=
#routeShapePath=!
#countyBoundaryPath=!
#censusTractPath=!
#clientLocationPath = !
#clientFileName =!
#routesPath =!
#routeFileName = !

#outputNearTable=
#radiusUnits=
#searchRadius
#numberOfNearest
#queryTableName

#IF MANUAL ENTRY IS CHOSE, ONCE ABOVE VARIABLES ARE MANUALY DEFINED SET routeAndClientAndGDBAssigned to 'Y' so that rest of code runs
#routeAndClientAndGDBAssigned = 'Y'
```

Figure 2: Manual File Name and Path Entry Option

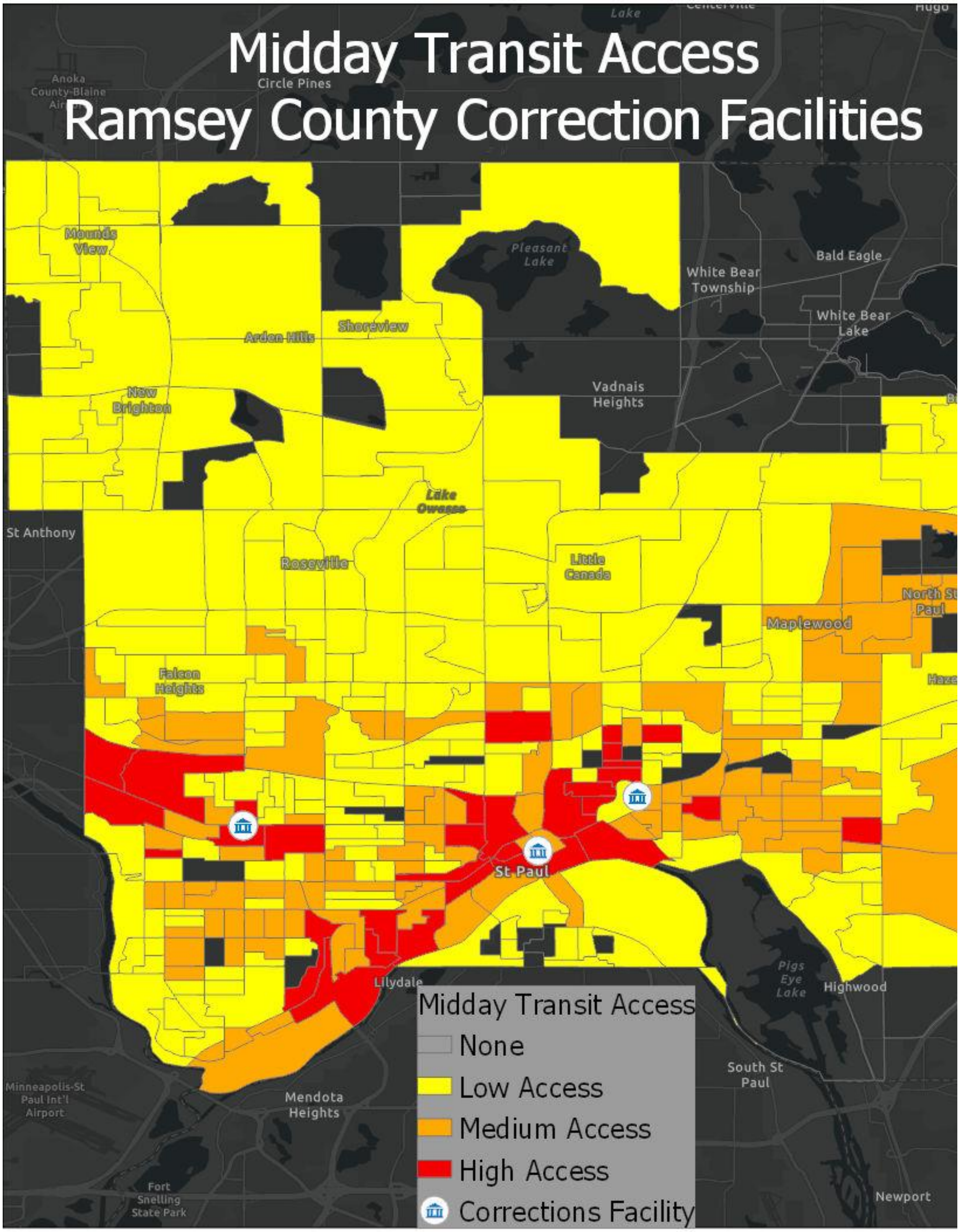
3. **CONVERT DATA TO SHAPEFILES**
    - Based on the specified input files, all are converted to shape files for visualization in ArcGIS.
    - Client Location and Route Geometry shape files are clipped using the county boundary shapefile such that the only features that are kept are those which exist within the county
4. **CREATE NEAR TABLE & RANK ROUTES**
    - A near table is created that lists all transit stops (and thus routes) that occur within 0.25 miles of an individual client
    - By looking at the routes near each individual, the total number of client within a quarter mile of a route is determined and then the routes are ranked in order of how many clients they are near
    - This RouteProximity ranking will be one of the two factors in calculating transit accessibility
5. **CALCULATE FINAL FIELDS**
    - The final component of accessibility that needs to be accounted for is the frequency of each route.
    - The frequency is extracted from publicly available route headway data
    - The final transit access measure is calculated by multiplying this frequency attribute by the corresponding Route Proximity as calculated in step 4. This final value is saved as
      "routeProxandFreq"
6. **IMPORT TO ARCGIS AND LAST SYMBOLOGY ADJUSTMENTS**
    - Step 5 is where the python code terminates
    - After the termination of the code, the clipped route data needs to be added to an ArcGIS map
    - Visualization can be determined by the user but the way it was analyzed for this report was as shown in Figure 3.

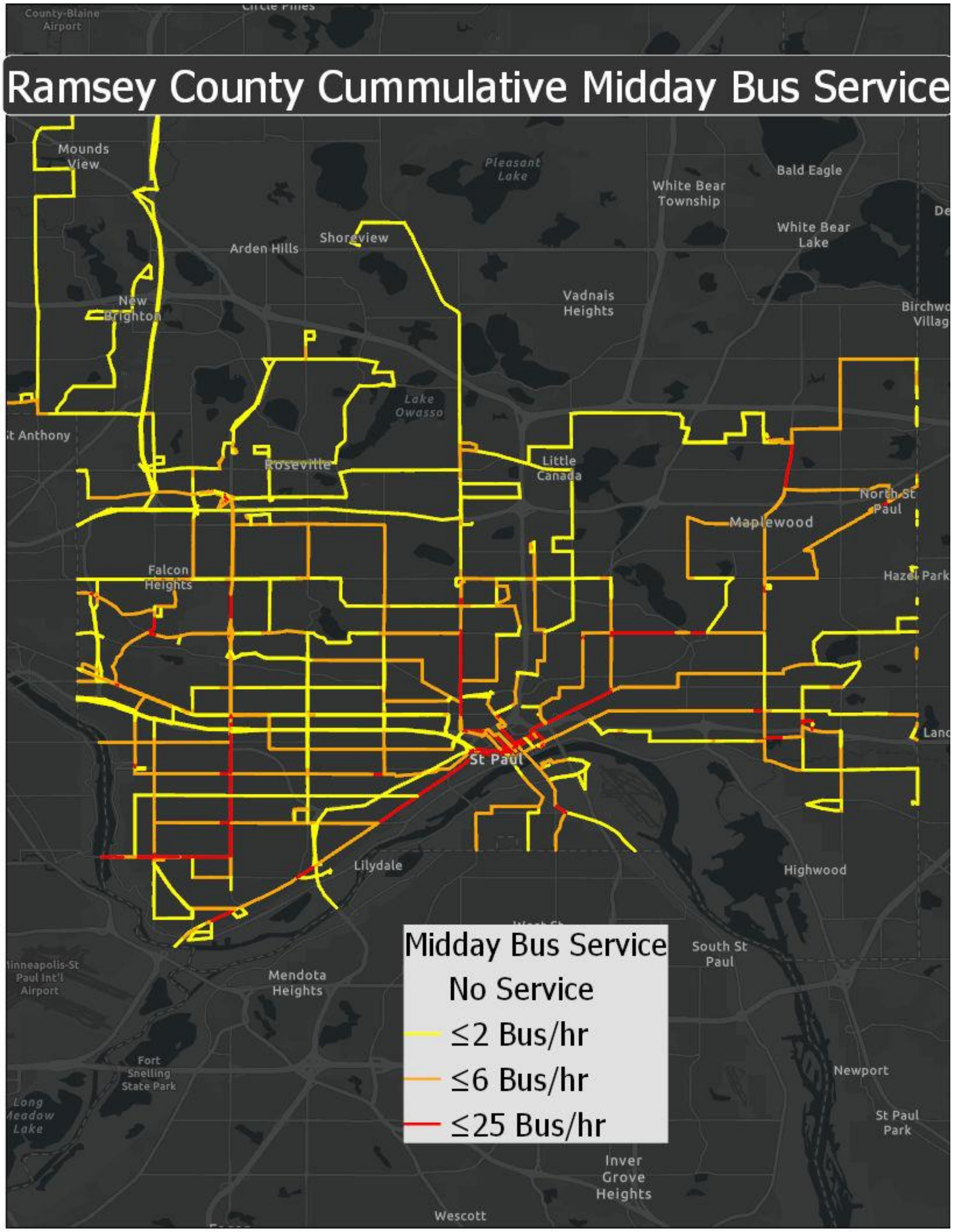
```
#   7. Process of displaying data with files created above

## 1. Import Data From GDB FOLDER
#       a.Clipped Client Shape File (ex. Clipcorrec)
#       b.bdry_commissionerdist2012.shp
#       c.ClipRouteColor.shp
#       d.ClipCensus.shp
#
## 2. Set extent to be centered on Ramsey County
#
## 3. Create Summary of Route Info per Census Tract (Use Summarize Within Tool)
#       a.InputPolygons==ClipCensus.shp
#       b.InputSummaryFeatures==ClipRouteColor from above
#       c.OutputFeatureClass='NAME YOU CHOOSE'
#       d.'Keep all polygons'->UnCheck
#       e.SummaryFields
#               ['routeProxandFreq,'MEAN']]
#       f.'Add Shape Summary'-> Check
#       f.Shape Unit=Miles
#       g.GroupField='Line_id' VERY IMPORTANT TO HAVE THIS
#       h.OutputGroupedTable='ENTER NAME HERE'
#
## 4. Use GroupedField Table From Previous Step TO Calculate The Sum of Average Values Per Census (Use Summary Statistics)
#       a.Input Table=Line_Id summary table from 3g above
#       b.OutputTable='Whatever Name'
#       c.Statistic Fields=sum of Mean_MID_HDWY and sum of Mean_MID_Count and sum of Mean_WeightedProximityFieldName
#       d.Case Field= Join ID
#
## 5. JOIN Summary_Stats Table from 4b. back to SumarizedWithin File from 3c
#       a.LayerName=ClipCensusSummarizeWithinFile (From Step 3c. above)
#       b.Input Join Field== JOIN ID
#       c.JoinTable='Line ID Summary Stats table name from 4b.
#       d.Output Join Field== JOIN ID
#
## 6. Change symbology of census tracts in summarize within to be graduated based on sum of mean weightedProximityField
#       a. Use 4 Categories
#           i. NONE
#           ii. LOW ACCESS
#           iii.MEDIUM ACCESS
#           iv. HIGH ACCESS
#
## 7. Use Heat Map with 15 nearest points to vizualize client locations
```

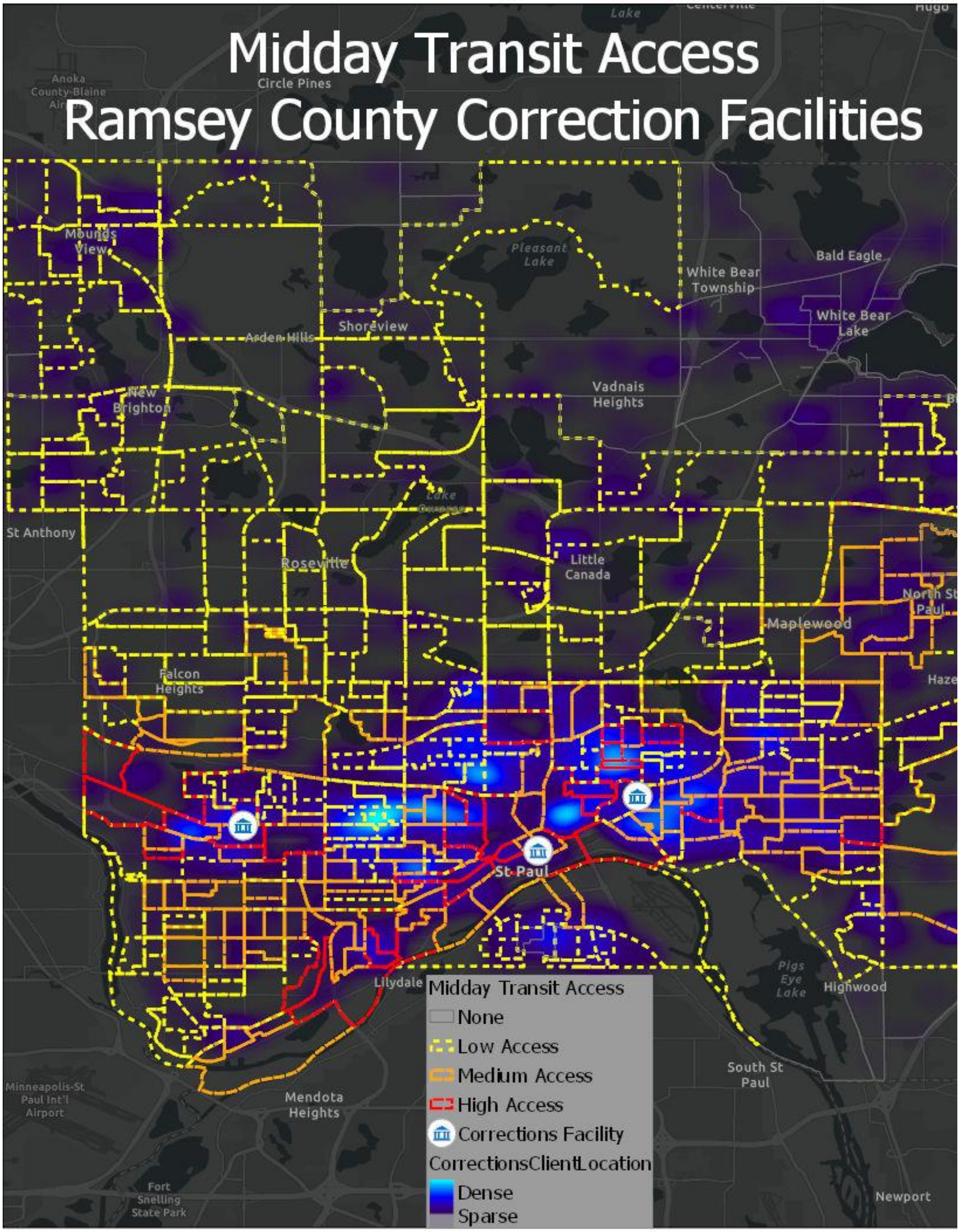
Figure 3: Final Steps Taken Within ArcGIS Using Code Output

Sample code output maps are reproduced on the following pages. For more detail regarding the code specifics please see comments contained within code files.

SAMPLE FINAL OUTPUT




Ramsey County Cummulative Midday Bus Service
Midday Bus Service
No Service
≤2 Bus/hr
≤6 Bus/hr
≤25 Bus/hr
County-Blaine Airport
Mounds View
Pleasant Lake
White Bear Township
Bald Eagle
White Bear Lake
Arden Hills
Shoreview
Vadnais Heights
New Brighton
Birchwood Village
Lake Owasso
St Anthony
Roseville
Little Canada
North St Paul
Maplewood
Falcon Heights
Hazel Park
St Paul
Lilydale
Highwood
South St Paul
Minneapolis-St Paul Int'l Airport
Mendota Heights
Fort Snelling State Park
Long Meadow Lake
Newport
St Paul Park
Inver Grove Heights
Wescott


Midday Transit Access
Ramsey County Correction Facilities
Circle Pines
Mounds View
Pleasant Lake
White Bear Township
Bald Eagle
White Bear Lake
Shoreview
Arden Hills
New Brighton
Vadnais Heights
St Anthony
Roseville
Little Canada
North St Paul
Maplewood
Falcon Heights
St Paul
Lilydale
Pigs Eye Lake
Highwood
South St Paul
Mendota Heights
Minneapolis-St Paul Int'l Airport
Fort Snelling State Park
Newport
Midday Transit Access
None
Low Access
Medium Access
High Access
Corrections Facility
CorrectionsClientLocation
Dense
Sparse